Tienda de comics Ultimis

Juan Diego García Vargas

Diego Alejandro Rosazza Merida

Versión 1.1

Cochabamba, 18 de 09 de 2018

Registro Histórico de Cambios y Revisiones

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** | **Autor Responsable** | **Aprobado** |
| 05/09/2018 | 0.0 | Presentación del tema | Juan Diego García | 05/09/2018 |
| 05/09/2018 | 0.0 | Presentación de los Objetivos | Diego Rosazza | 05/09/2018 |
| 14/09/2018 | 0.3 | Presentación del Diagrama de Casos de Uso | Diego Rosazza | 14/09/2018 |
| 05/09/2018 | 0.0 | Presentación del Modelo de Datos | Juan Diego García | 05/09/2018 |
| 14/09/2018 | 0.4 | Presentación del Diagrama de Clases | Diego Rosazza | 14/09/2018 |
| 07/09/2018 | 0.0 | Subida Primeras relaciones funcionales | Juan Diego García | 07/09/2018 |
| 0709/2018 | 0.1 | Cambiados accesos en cascade en dbforge | Juan Diego García | 07/09/2018 |
| 10/09/2018 | 0.2 | Creados métodos y entidades restantes | Diego Rosazza | 11/09/2018 |
| 11/09/2018 | 0.3 | Presentación del Primer Prototipo | Diego Rosazza | 11/09/2018 |
| 12/09/2018 | 0.4 | Reestructuración de todos los menús | Juan Diego García | 12/09/2018 |
| 17/09/2018 | 1.0 | Añadidos métodos faltantes | Juan Diego García | 17/09/2018 |
| 18/09/2018 | 1.1 | Realizada restructuración final | Juan Diego García | 18/09/2018 |
| 18/09/2018 | 1.1 | Presentación del Proyecto Final | Diego Rosazza | 18/09/2018 |

Tabla de Contenidos

1. Introducción 4

2. Análisis de Requerimientos 4

2.1 Requerimiento *«Gerente»* 4

2.1.1 Requerimiento *«Manejar Datos Comics»*  4

2.1.2 Requerimiento *«Manejar Datos Dibujante» 4*

*2.1.3 Requerimiento «Manejar Datos Distribuidora» 5*

*2.1.4 Requerimiento «Manejar Datos Guionista» 5*

*2.1.5 Requerimiento «Manejar Datos Personaje» 5*

*2.1.6 Requerimiento «Manejar Datos Condicion Especial» 5*

*2.1.7 Requerimiento «Manejar Datos Detalles Compra/Venta» 5*

*2.1.8 Requerimiento «Manejar Datos de Compra y Venta» 5*

*2.1.9 Requerimiento «Manejar Datos Facturas» 5*

*2.1.10 Requerimiento «Realizar Consultas Compra/venta» 5*

*2.1.11 Requerimiento «Realizar Consultas Ganancias» 5*

*2.1 Requerimiento «Cliente» 6*

*2.2.1Requerimiento «Realizar Consultas Tienda» 6*

3. Objetivo General 6

4. Objetivos Específicos 6

5. Límites y Alcances 6

6. Marco Teórico 7

7. Diagramas de Casos de Uso 8

8. Modelo de Datos 9

9. Diagrama de Paquetes 9

10. Pruebas y Validación 11

11. Conclusiones 11

12. Recomendaciones 12

13. Bibliografía 12

Informe Final del Proyecto

# Introducción

Los comics son una de las formas de narrar una historia más conocidos en estos días, los cuales suelen tratar las historias de héroes y grandes batallas, las cuales ocurren en nuestro mismo espacio y tiempo, generalmente reflejando la realidad social que se vive en una época, por ende, son complejos y variados.

Por ende, una tienda de comics debe ser capaz de mantener un registro completo y detallado de los comics, junto con la información necesaria para poder manejar una tienda de manera eficiente.

Es necesario crear una base de datos capaz de manejar el inventario y compra/venta de comics.

Cada comic tiene una serie diversa de datos a los que se les puede hacer referencia, muchos de estos compartidos por más de un comic, como ser los personajes que tiene, su distribuidora, los creadores del comic, los dibujantes del comic, su precio de compra, su precio de venta y si es de edición especial o cuenta con alguna nominación “especial”.

De los detalles de la distribuidora se tiene, los precios de venta, sus datos de venta y factura.

De los detalles de venta al cliente se tienen los detalles de ventas históricos más sus facturas.

Se debe poder ver, agregar, editar y eliminar cada una de estas clases, exceptuando las de venta y compra que no podrán ser editadas ni eliminadas

La base de datos es el gerente de la tienda, el cual recopila los datos de todos los movimientos hechos en cualquier sector y las características que engloba, teniendo los comics todos los datos para poder organizarlos y venderlos ya en la repisa, y siendo solo necesario que los empleados realicen el cobro de forma manual y directa según la información del comic.

Además, se tiene la opción de que tanto empleados como clientes puedan ver las características respecto a un comic, el cual no se extiende más allá de los precios de venta y no muestra los precios de compra, pero puede profundizar en otros datos relacionados

# Análisis de Requerimientos

En la base de datos a crearse, se tienen 2 usuarios distintos que pueden usar el sistema, uno es el gerente, el cual tiene control administrativo de toda la tienda ay cada uno de sus detalles, pudiendo modificar, eliminar, mostrar o actualizar la mayoría, exceptuando las de carácter financiero, con las que goza de una libertad limitada para manejar esos datos. El otro es el cliente, el cual puede ver todos los valores

## Gerente

### **Requerimiento “Manejar datos de Comics”**

* Poder crear, editar, listar y eliminar comics.
* Que no se repita el código de un comic.
* Ninguna

### **Requerimiento “Manejar datos de dibujantes”**

* Poder crear, editar, listar y eliminar dibujantes.
* Que no se repita el código de un dibujante.
* ninguna

### **Requerimiento “Manejar datos de distribuidora”**

* Poder crear, editar, listar y eliminar distribuidoras.
* Que no se repita el código de una distribuidora.
* ninguna

### **Requerimiento “Manejar datos de guionista”**

* Poder crear, editar, listar y eliminar guionistas.
* Que no se repita el código de un guionista.
* ninguna

### **Requerimiento “Manejar datos de Personajes”**

* Poder crear, editar, listar y eliminar Personajes.
* Que no se repita el código de un personaje.
* ninguna

### **Requerimiento “Manejar datos de Condición Especial”**

* Poder crear, editar, listar y eliminar Condiciones Especiales.
* Que no se repita el código de una Condición Especial.
* Ninguna

### **Requerimiento “Manejar datos de detalles de compra y venta”**

* Poder crear, editar, listar y eliminar detalles relacionados a la Compra/venta.
* Que no se repita el código de una Compra/Venta o que no exista una venta o comic asociado al detalle.
* Ninguna

### **Requerimiento “Manejar datos de compra y venta”**

* Poder crear, editar y listar Compras/ventas.
* Que no se repita el código de una Compra/Venta.
* ninguna

### **Requerimiento “Manejar datos de facturas”**

* Poder crear, editar, listar y anular detalles relacionados a las facturas de Compra/venta (Una factura anulada anula la compra/venta asociada).
* Que no se repita el código de una Compra/Venta.
* ninguna

### **Requerimiento “Realizar consultas de compras/ventas”**

* Realizar consultas según los datos deseados de compras o ventas.
* ninguna
* ninguna

### **Requerimiento “Realizar consultas de ganancias”**

* Poder ver las ganancias o pérdidas referentes a un comic y/o característica.
* ninguna
* ninguna

## Cliente

### **Requerimiento “Realizar consultas de la tienda”**

* Poder ver todos los detalles de la tienda no financieros que desee como los guionistas, etc.
* ninguna
* ninguna

# Objetivo General

Se pretende tener un registro fiel y completo de una tienda de comics, capaz de ser usado por los clientes y el gerente de la tienda de forma independiente.

El gerente podrá controlar la totalidad de los datos internos en la tienda y además ver los que se encuentran en sistema y el cliente podrá informarse acerca de los productos que desea conseguir.

# Objetivos Específicos

Ser capaces de introducir, mostrar, modificar y eliminar datos en todas las secciones que lo requieran, teniendo de esta forma el esqueleto de la base de datos.

Crear métodos de consultas avanzados para obtener los datos deseados respecto al manejo económico de la tienda, es decir datos monetarios de ingresos y egresos en ciertos comics, personajes, guionistas, etc.

Tener un menú organizado orientado a la funcionalidad de los diferentes métodos, en los cuales se dividirán los métodos administrativos de los financieros y consultas específicas, el solo podrá ser usado por el gerente, el cual deberá introducir un código especial para acceder al mismo, teniendo el cliente un menú simplificado que le permitirá navegar por la base de datos relacionada a características y comics que pudiese desear.

# Límites y Alcances

La aplicación desarrollada solo podrá funcionar dentro de eclipse, sin tener ninguna interfaz gráfica distinta la barra de comandos, esto se debe al proceso de desarrollo de la base de datos, el cual para ser funcional demora mucho tiempo, por ende, no fuimos capaces de elaborar una interfaz grafica.

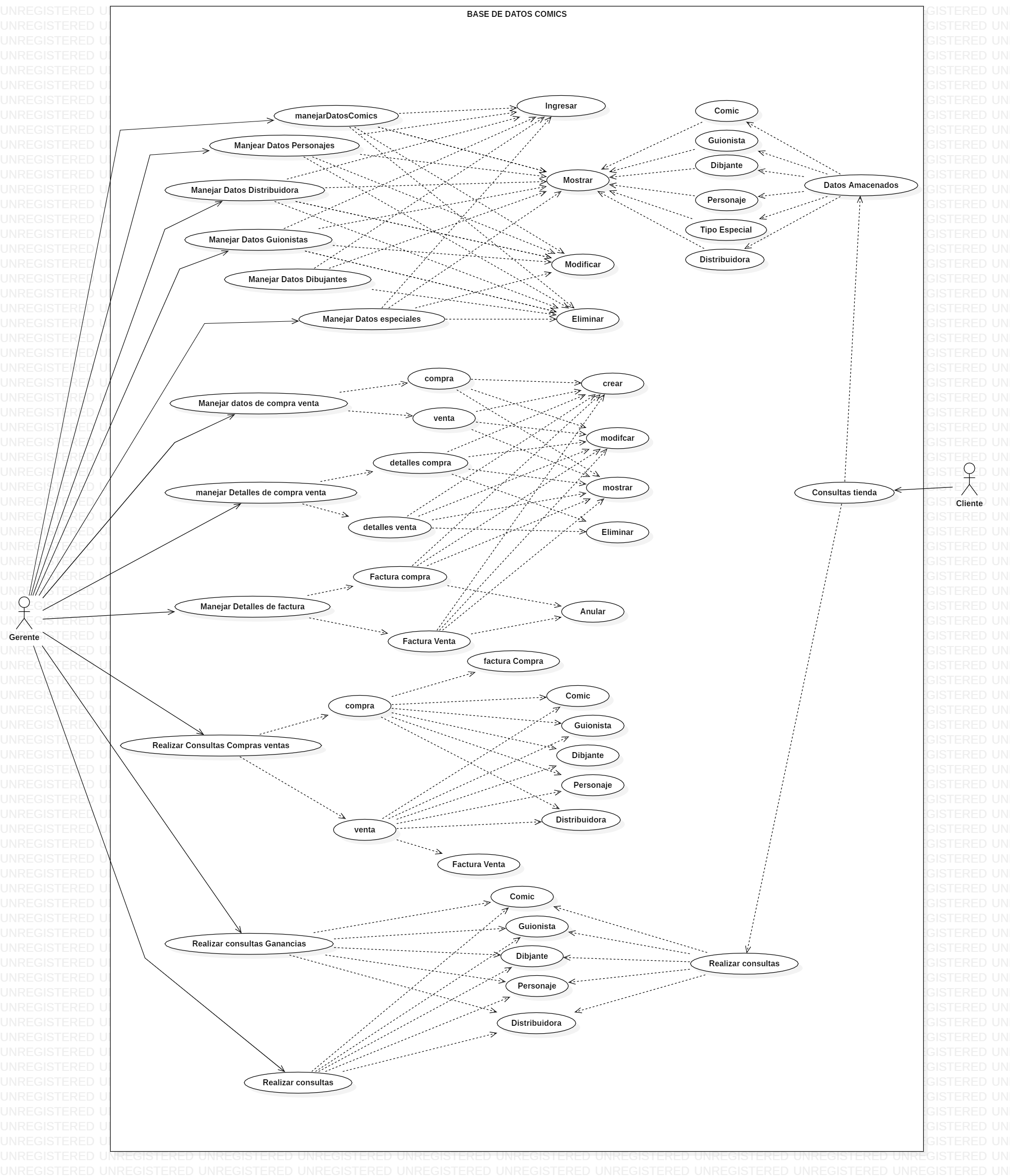
Se encontró que faltaba un detalle al final del desarrollo, el cual no permite mostrar de forma exacta cuantos comic quedan dentro de la tienda, esto se debe a que no consideramos una forma de almacenar los comics disponibles dentro de la base de datos, aun así, se vieron 2 posibles formas de solución arlo que no se llevaron a cabo debido al factor tiempo, una siendo realizar una comparación entre la cantidad de comics comprados y vendidos con un mismo código para obtener el total restante, y la otra almacenando el dato de forma interna dentro de comic, sin embargo, al enfocarnos en solucionar los otros problemas y completar los otros requerimientos esto no se pudo realizar.

Debido a la forma en que se crearon las relaciones en los tipos especiales, si se quisiera tener un mismo comic con otra característica especial sería necesario tener otro código de barras para diferenciar los diferentes comics, con tal de tomarlos como entidades distintas. Cabe resaltar que la relación presente en el modelo de dbforge permite que un comic tenga varias características especiales, sin embargo, nos percatamos por el final que esto resultaría en que un comic, por especial que fuese, se vendería al mismo precio de uno regular, razón por la cual se creó esta limitante.

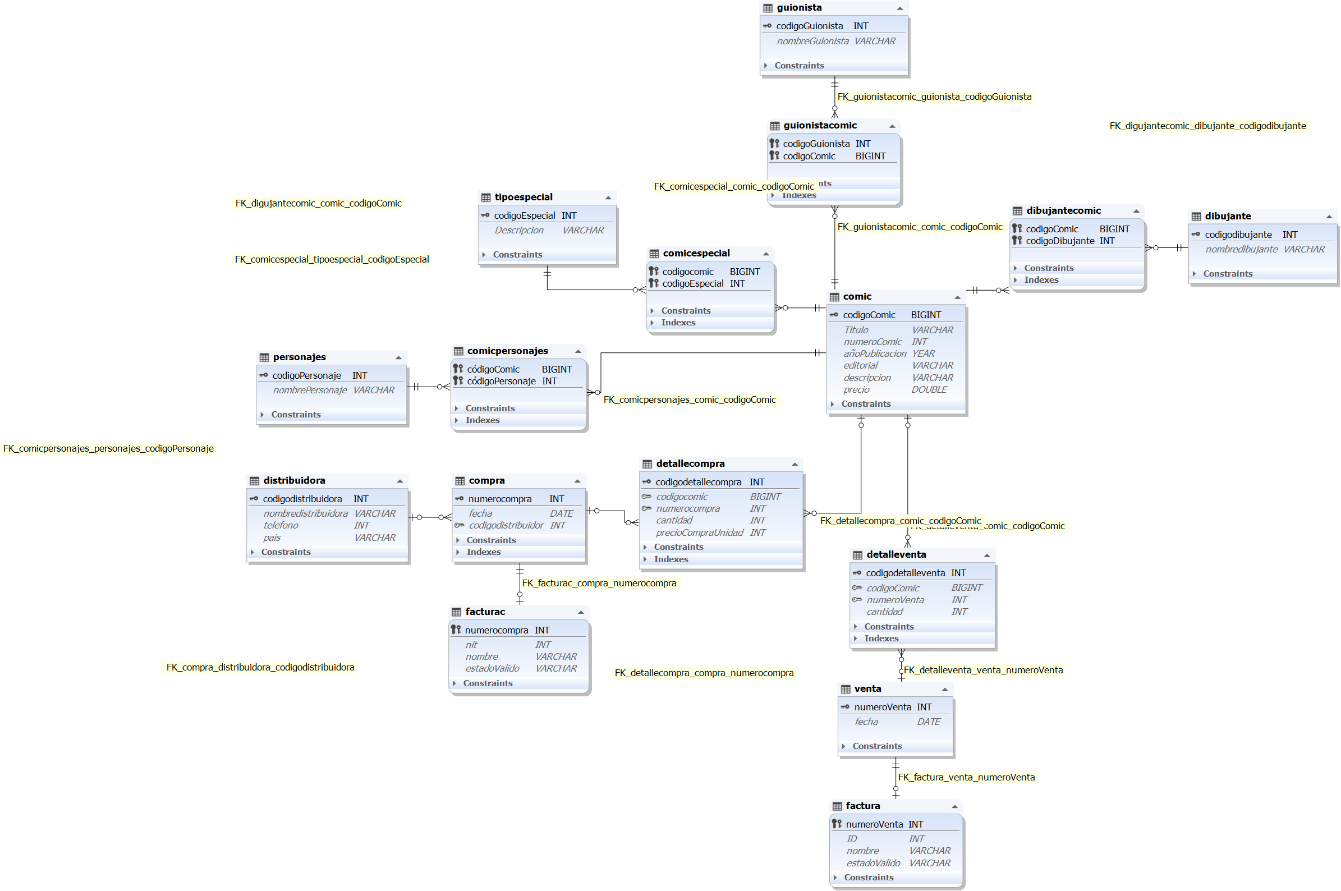
# Marco Teórico

Fue necesario conocer el funcionamiento de java y el entorno de desarrollo eclipse, además del conocimiento interno de una tienda de comics, las características que trae un comic y las relaciones entre sus características.

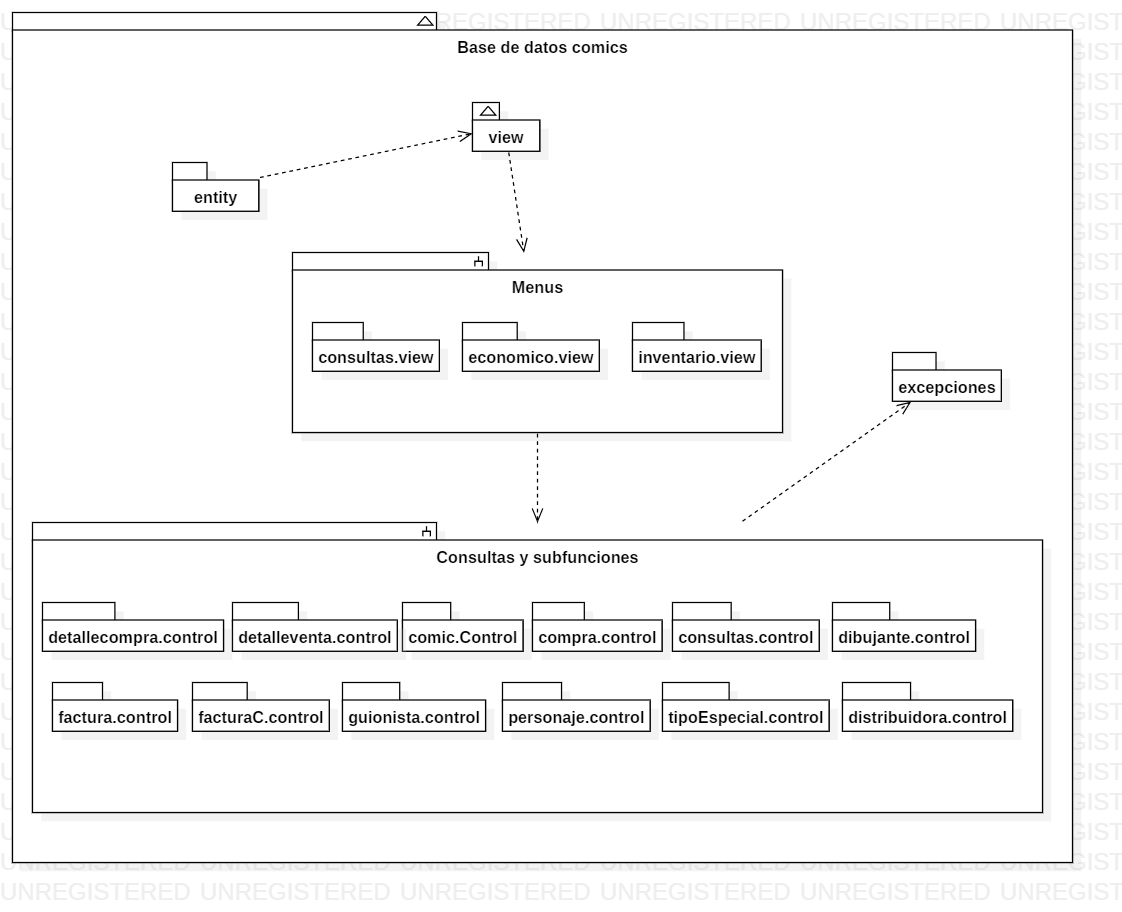
# Diagrama de Casos de Uso



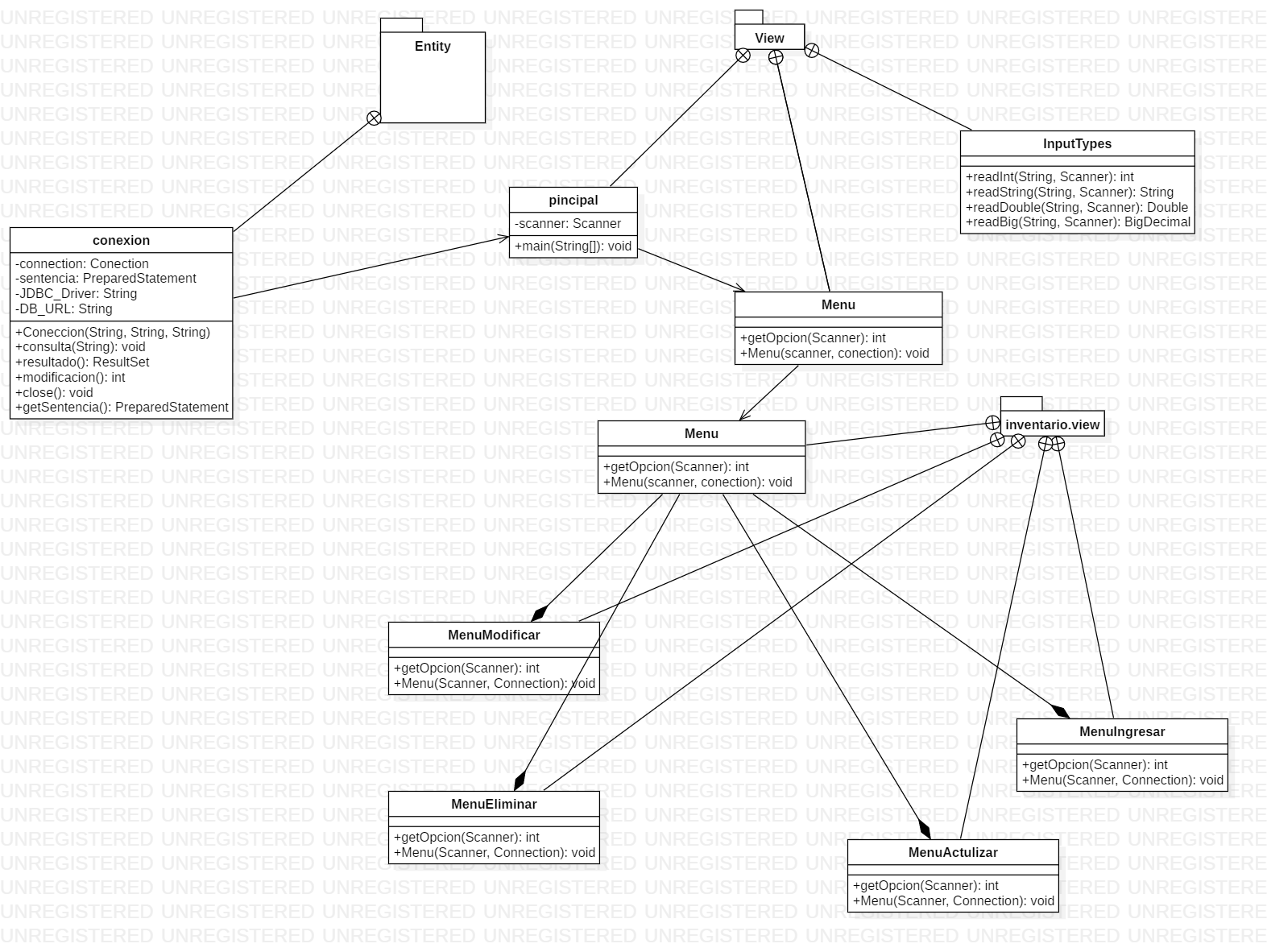
# Modelo de Datos



# Diagrama de Paquetes



Se crearon varios paquetes para poder identificar de mejor manera las funcionalidades de cada grupo identificad en el modelo de dbforge, es decir, las funcionalidades de cada tabla en la que se desean realizar consultas, crear datos, listar contenido, eliminar contenido y actualizar contenido, por ende, la mayoría de las funciones se repiten en cada paquete, cuya única diferencia recae en detalles en la orden de sql que se manda, para que esta se pueda usar en la tabla que desea, dicho esto, la relación entre clases depende de la relación entre paquetes que existe, ya que por ejemplo, el paquete inventario.view tiene internamente 5 clases, los cuales son menús, uno para elegir que acción de inventario se desea realizar y derivar al usuario al submenú correspondiente, el cual hace referencia a una clase correspondiente en otro paquete completamente distinto. Esto suena críptico, por ende se tiene la siguiente imagen para dar mas claridad a la forma en que todo esta estructurado por dentro.



Esta situación se repite para cada sub menú relacionado a una funcionalidad diferente, como el ver lo datos económicos o las consultas, razón por la cual no se realizara un diagrama completo de las clases, al ser estas repetitivas y dependientes de la relación entre paquetes.

Pero basta con saber que cada paquete tiene almacenada clases que hacen referencia directa a su nombre, es decir, todos los views tienen internamente menús para dirigir al usuario entre las opciones operacionales de los paquetes control, los cuales se encargan de realizar las consultas en sql, y dependiendo de que se pida de ellos, estos almacenaran clases distintas para cada consulta, dicho esto, todos los paquetes que tengan relaciones con inventario.view tienen internamente clases de eliminar, mostrar, crear y modificar, los paquetes relacionados con económico.view tienen métodos para crear, eliminar, mostrar y modificar, en los cuales las facturas reemplazan el eliminar por un anular, y los otros paquetes realizan consultas de ventas o compras de valores específicos, por ultimo, los paquetes relacionados a consultas.view manejan métodos para realizar consultas de cantidad y características deseadas de comics o características del mismo, a la vez de tener la ganancia neta según una característica deseada.

# Pruebas y Validación

Al principio cuando creamos la base de datos en dbforge realizamos consultas internas para ver que las relaciones entre tablas funcionen de la forma correcta, posteriormente empezamos a replicar los métodos en java siguiendo el ejemplo dado por el docente referente a Tienda, esto corresponde a la versión 0.0.

Posteriormente cambiamos los accesos de las entidades respecto a los métodos DELETE y UPDATE a ON CASCADE con tal que al borrar o actualizar algún dato de alguna entidad, todas las entidades que utilicen los datos actualizados se modifiquen de manera simultánea, esto corresponde a la versión 0.1.

Al momento de crear todas las entidades y sus funcionalidades básicas de añadir, eliminar, listar y actualizar se crearon varios paquetes y clases, lo que corresponde a las versiones 0.2 y 0.3.

A medida que se continuó con el proyecto se tuvo la necesidad de cambiar todos los menús para hacerlo mas accesible y amigable al usuario, en ese momento se dividieron los menús por funciones y no por sus entidades, como estaba organizado anteriormente, este cambio corresponde a la versión 0.4.

Una vez se tuvo los menús finales, se crearon las consultas faltantes, que están relacionadas a las compras, ventas y ganancias, esta ya fue la primera versión funcional completa según los requerimientos a cumplir, la cual corresponde a la versión 1.0.

Por ultimo, se creo un menú individual para el cliente, reciclando los métodos de gerente que fuesen necesarios, cumpliendo finalmente todos los requerimientos mas el acceso al cliente que se teorizo al principio, esto corresponde a la versión 1.1.

Para cada nueva consulta en java creada se realizó un a prueba de ejecución del comando creado desde java, por lo cual existe constancia de que el código creado funciona en su totalidad.

# Conclusiones

La base de datos creada es capaz de satisfacer todos los requerimientos encontrados, en el proceso se tuvo que reestructurar el sistema interno del dbforge para tener métodos de update y delete que fuesen sencillos de ejecutar, posteriormente, tuvimos problemas con el manejo de github, razón por la cual se concluyó que la mejor manera de evitarlos era turnarse al hacer cambios y commits, de forma tal que solo se usara la branch master.

Respecto a java, tuvimos problemas de manejo de errores, cuando una excepción no podía ser retenida de forma adecuada, esto requirió crearse una excepción propia para eludir el problema.

Por falta de tiempo, la interfaz del programa elegida fue la barra de comandos y no algún apartado grafico como JavaFX o JavaSwing, debido a esto al momento de mostrar datos, estos se muestran de una manera irregular.

# Recomendaciones

Una limitación de la base de datos es el hecho de que un comic tiene un solo precio de venta a pesar de tener varias ediciones especiales, sin embargo, este problema se puede evitar, principalmente porque regularmente ediciones distintas tienen diferentes códigos de barras, a futuro se podría ver una manera de introducir el precio añadido por una condición especial, de forma que un mismo comic cuente con varias condiciones especiales y un precio distinto para cada una.

El programa realiza las relaciones con los guionistas, dibujantes, personajes, etc. al momento de ser creada, debido a esto es necesario crear los datos necesarios previos a ingresar un comic nuevo al sistema, en un futuro si se continuase con el desarrollo se podría hacer que al momento de introducir un dato referenciado inexistente se dé la opción de crearlo en vez de lanzar un mensaje de error y pedir un dato valido.

SI se puede continuar el proyecto, se podría seguir principalmente con el desarrollo del apartado gráfico, haciendo posible ver la base de datos de una manera ordenada y cómoda, además de que se podría añadir las caratulas de los comics.

Por último, de continuarse el desarrollo de la base de datos sería necesario tener un método que nos permita mostrar la cantidad de comics disponibles para la venta, ya se tienen 2 métodos posibles para implementarlo, pero es necesario contar con más tiempo del que poseemos actualmente para completarlo.

# Bibliografía

Barrero, L. (2013). *Estructura de Datos.* Cochabamba: UPB.

MariaDB. (10 de 09 de 2018). *MariaDB Knowledge Base*. Obtenido de https://mariadb.com/kb/es/